

## 2 Wann brauchen Sie eine EAI-Lösung?

Wie Sie noch sehen werden, ist der Begriff EAI nicht gerade präzise definiert. Wenn Sie allerdings vom Verkäufer eines EAI-Tools angesprochen werden, wird der Ihnen klar machen, dass Sie genau sein Tool benötigen. Oft wird er das tun, ohne wirklich nach Ihrem Anwendungsfall zu fragen – one tool fits all.

Dieses Kapitel gibt Ihnen zunächst einen ersten begrifflichen Überblick über EAI. Sie werden dann die Trends kennen lernen, die den Bedarf nach EAI-Lösungen wachsen lassen.

Dann erfahren Sie Näheres zu vier häufigen Anwendungsfällen für EAI-Lösungen. Es sind dies:

*EAI-Anwendungsfälle*

- ❑ Multichannel-Architekturen,
- ❑ Application to Application (A2A) Integration,
- ❑ Geschäftsprozessintegration mit Workflow und
- ❑ Anwendungen aus dem Bereich Trade Rooms.

Das Kapitel stellt die rein durch das Geschäft verursachten Anwendungsfälle für diese Technologien dar. Über die Technologien selbst lesen Sie zunächst noch wenig. Es gibt sicher noch mehr Anwendungsfälle. Die hier aufgeführten und beschriebenen treten jedoch sehr häufig auf.

### 2.1 Was ist EAI?

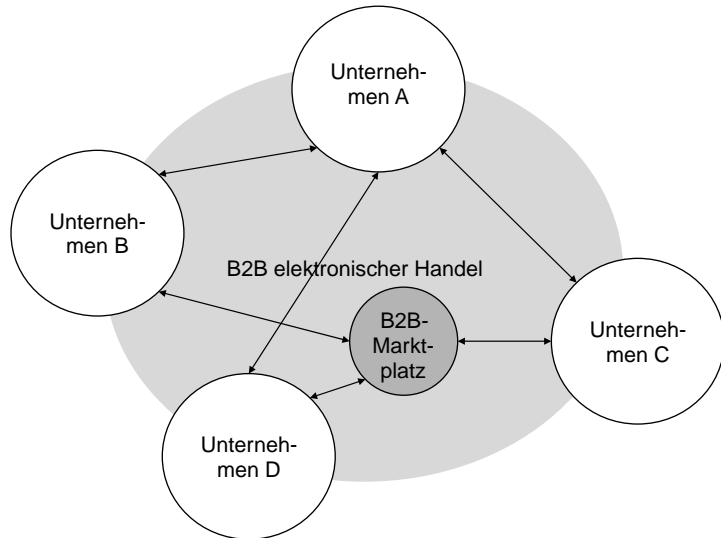
EAI heißt Enterprise Application Integration. EAI-Produkte sind Software, die es erlaubt, die Anwendungen eines Unternehmens zu integrieren. Der Begriff wurde geprägt durch das Bemühen, viele Anwendungen, die nicht für eine Zusammenarbeit entworfen wurden und auch nur Teilaufgaben von Geschäftsprozessen abdecken, dazu zu bringen, in einheitlichen Geschäftsprozessen zusammenzuspielen. Es geht also darum, heterogene Anwendungen eines Unternehmens so zu integrieren, dass sie sich möglichst so verhalten, als wären sie von Anfang an dafür entworfen worden, die aktuellen Geschäftsprozesse eines Unternehmens zu unterstützen. Abschnitt 2.3.3 beleuchtet dieses spezielle Benutzungsszenario von EAI später noch näher.

*Begriff*

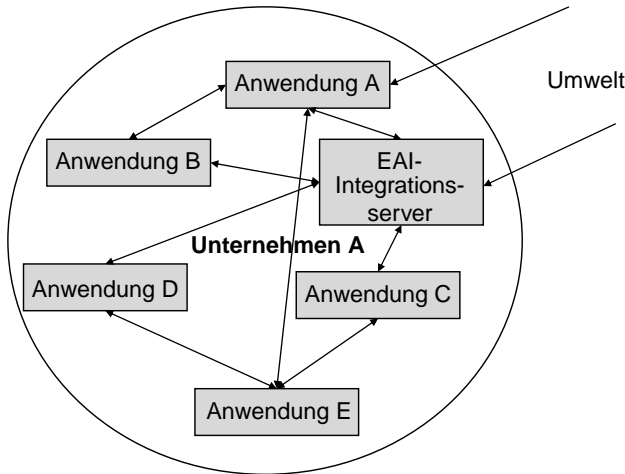
Da die Werkzeuge, die wir in diesem Buch vor allem erläutern, nämlich so genannte EAI-Integrationsserver, auch in anderen Berei-

chen Verwendung finden, macht es hier Sinn, EAI von so genanntem B2B-E-Commerce kurz abzugrenzen.

**Abbildung 1:**  
B2B-E-Commerce  
Die Pfeile stellen den  
Informationsaustausch  
zwischen den Partnern  
dar.



**B2B** **B2B-E-Commerce** rationalisiert den Handel zwischen Unternehmen. Mittels Geschäftsprozess-Reengineering gestalteten viele Firmen in den 90er Jahren ihre internen Prozesse neu. Dabei beschleunigten sie vor allem den Fluss von Gütern. Heute machen die Unternehmen den Fluss von Informationen zwischen Geschäftspartnern schneller. B2B-E-Commerce wird also vor allem eingesetzt, um schneller und mittels elektronischem Datenaustausch Geschäfte zwischen Unternehmen zu vereinbaren und abzuwickeln. Das ist nicht neu. Die Versuche gibt es mit EDI schon mehr als 20 Jahre. Nur durch neue Technologien, wie vor allem das Internet und die dort verwendeten Technologien wie XML, SOAP, Webservices und auch Integrationsserver, erhalten diese Bestrebungen neuen Schwung und die Unternehmen hoffen, bei diesem Anlauf mehr zu erreichen. Zum Teil wurde auch schon in bestimmten Industrien viel erreicht, wie die Geschichte von RosettaNet zeigt.



**Abbildung 2: EAI**  
Die Pfeile stellen wieder den Informationsaustausch dar. Die Unternehmensumwelt kann entweder direkt mit einer Anwendung (A) kommunizieren oder die Kommunikation mit der Unternehmensumwelt wird über einen EAI-Integrationsserver konzentriert abgewickelt. Dieser kann auch als Hub & Spoke-Verteiler für Informationsaustausch im Unternehmen eingesetzt werden.

EAI (Enterprise Application Integration) beschleunigt und rationalisiert die Informationsflüsse innerhalb eines Unternehmens. Wenn man sich Abbildung 2 ansieht, wird man feststellen, dass sie strukturell identisch zu Abbildung 1 ist. Die Unternehmen wurden durch die einzelnen Anwendungen eines Unternehmens ersetzt, die miteinander kommunizieren müssen, und der B2B-Marktplatz durch den EAI-Integrationsserver. Damit zielt EAI auf die Rationalisierung innerhalb des Unternehmens.

EAI

Es ist einleuchtend, dass die Werkzeuge, die man für EAI und B2B-E-Commerce verwendet, strukturell ähnlich sind. Dieses Buch beschäftigt sich allerdings schwerpunktmäßig mit EAI. B2B-E-Commerce werden wir in einigen Abschnitten streifen, vor allem dann, wenn die Strategie von Microsoft erläutert wird. Deren EAI-Tool, der BizTalk-Server, stammt eher aus dem Bereich B2B-E-Commerce und es macht daher auch Sinn, dieses Gebiet im Überblick zu erläutern, wenn man ein gutes Verständnis für EAI erreichen will.

Werkzeuge

Die Werkzeuge, die für EAI entwickelt wurden, kann man für viele Anwendungsfälle in Unternehmen verwenden, so zum Beispiel um so genannte Multichannel-Architekturen aufzubauen (siehe dazu Abschnitt 2.3.1) oder um generell die Kommunikation zwischen Anwendungen eines Unternehmens einfacher und billiger zu gestalten (siehe A2A-Kommunikation, Abschnitt 2.3.2).

Das Akronym EAI ist zu einem Sammelbegriff für viele Kategorien von Softwarelösungen geworden, die sämtlich etwas mit der Integration heterogener Lösungen innerhalb eines Unternehmens zu tun haben. Trotzdem unterscheiden sich aber die optimalen Lösungen immer

noch sehr stark, je nachdem, um welche Art von Problem es sich handelt – auch wenn sie alle unter den Oberbegriff EAI-Lösungen fallen.

ENS

Um abzugrenzen, mit welcher Art von Softwareprodukten und -lösungen wir uns hier befassen wollen, macht es Sinn, noch ein weiteres Akronym einzuführen. Das Akronym ENS steht für Enterprise Nervous System. Gartner Research hat diesen Begriff eingeführt für die Summe aus:

- ❑ Integration Brokern – das sind Sternverteiler, die für die Kommunikation von Anwendungen genutzt werden, um über Adapter mit vielen Systemen kommunizieren zu können,
- ❑ Geschäftsprozessmanagementtools,
- ❑ Middleware zur Kommunikation,
- ❑ Webservern,
- ❑ Applikationsservern
- ❑ und Datenintegrations- und ETL<sup>1</sup>-Technologien.

Was enthält ein  
Anwendungsportfolio?

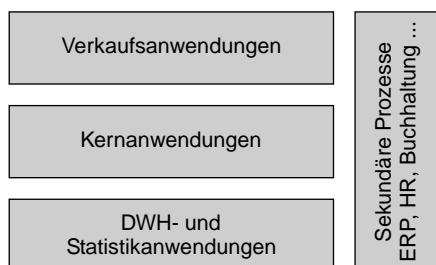
Bei den oben genannten Kategorien von Software handelt es sich um technische Software ohne fachlichen Inhalt. Wenn man sich nun das Anwendungs- und Softwareportfolio einer typischen Versicherung ansieht (siehe auch Abbildung 3), dann findet man dort:

- ❑ Anwendungen, die das Kerngeschäft abdecken. Es sind dies meist große, hochspezialisierte Hostanwendungen, die man auf dem freien Softwaremarkt von ERP-Herstellern noch nicht zukaufen kann. Bei Versicherungen ist dies zum Beispiel ein Vertragssystem für mehrere Millionen Lebensversicherungsverträge. Bei Banken stünde hier ein System für die Abwicklung von Zahlungsverkehr oder Aktienhandel mit Millionen von Transaktionen pro Woche.
- ❑ Spezialisierte Verkaufsanwendungen und Anbindungen an Vertriebspartner.
- ❑ ERP-Anwendungen. Also typischerweise standardisierte Anwendungen für sekundäre Geschäftsprozesse wie Hauptbuchhaltung, Controlling, Personalverwaltung und Ähnliches. Solche Anwendungen kann man von ERP-Herstellern, wie zum Beispiel SAP, Baan, Peoplesoft oder Oracle, zukaufen
- ❑ Data-Warehouse (DWH)- und Statistikanwendungen. Diese entstehen auf zugekaufter Basissoftware und decken den so genannten dispositiven Bereich ab. Für solche Lösungen haben sich Architek-

1. ETL ist ein Akronym für *extract, transform, load*. Man benutzt ETL-Tools, um Inhalte aus operativen Datenbanken zu extrahieren, diese zu transformieren und hinterher zum Beispiel in ein Data Warehouse zu laden. ETL Tools können oft auch für Datentransformationen im EAI-Bereich eingesetzt werden.

turstandards entwickelt. Siehe dazu zum Beispiel [Kimb+1998, Wiek1999].

Die Reihenfolge ist nicht zufällig so gewählt. Die Kernanwendungen waren meist zuerst vorhanden. Vielfach enthielten sie das, was heute unter den Kategorien ERP und DWH zugekauft wird, aber noch vor 25 Jahren ebenfalls selbst entwickelt wurde. Verkaufsanwendungen wurden später ab den 80er Jahren dazu gebaut, und zwar mit dem Aufkommen von mittlerer Datentechnik für dezentrale Einheiten, mit dem Aufkommen des PCs und mit der Verbreitung von Laptops.



**Abbildung 3:**

*Oberste Sicht auf das Anwendungsportfolio einer Versicherung*

Das Enterprise Nervous System, also die oben genannten Arten von technischer Software, wird zunehmend als Klebstoff und Schmiermittel benutzt, um Applikationen zusammenzufügen, kommunizieren zu lassen und um Inhalte für Geschäftspartner verfügbar zu machen – zum Beispiel im Web.

*ENS als GlueWare*

In diesem Buch beschäftigen wir uns näher mit den ersten drei Arten von Bestandteilen eines ENS, nämlich Integration Brokern, Geschäftsprozessmanagementtools und Middleware zur Kommunikation. Webserver und Applikationsserver sind hier nicht unser Thema. Mit ETL-Tools werden wir uns nur am Rande befassen, sofern sie auch EAI bei der Datentransformation unterstützen können.

## 2.2 Trends, die den Bedarf nach EAI wecken

In diesem Abschnitt soll der Frage nachgegangen werden, warum der Markt und der Bedarf für EAI-Lösungen derzeit dramatisch wächst. Der Satz:

*Internet*

Das Internet wird für immer die Art verändern, wie wir Geschäfte machen

ist zwar schon ein guter Einstieg, aber noch keine vollständige Erklärung, weil sich hinter dem Begriff eine große Menge von Technologien und Möglichkeiten verbergen, die man differenzierter betrachten muss.

### 2.2.1 Trend 1: Internet, B2B und neue Geschäftsprozesse

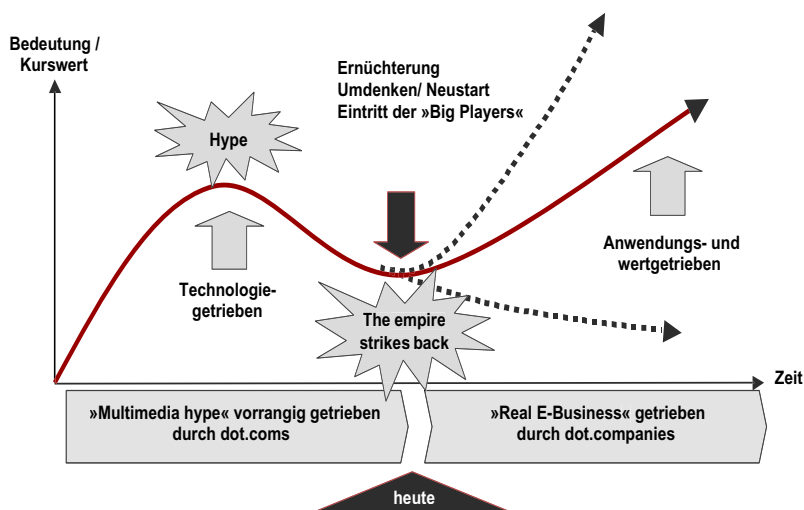
Der Internet-Hype der letzten Jahre vor dem Dotcom-Crash war geprägt durch so genannte B2C-Modelle, also Business-to-Customer-Modelle. Die Geschäftsmodelle, die dort primär implementiert wurden, waren (nach Musil et al. [Musil2000]) folgende:

- Verkehr verkaufen*     **Verkehr erzeugen und verkaufen:** Auf diesem Geschäftsmodell basieren Suchmaschinen, Verzeichnisdienste und zum Beispiel kostenfreie E-Mail-Provider und ähnliche Anbieter. Der Betreiber eines Internetauftrittes versucht, möglichst viel Verkehr auf seine Seiten zu ziehen, den Kunden dort möglichst lange zu halten, und erzielt Einnahmen über Werbung.
- Kundenorientierung*     **Kundenorientierung zum Beispiel im elektronischen Handel:** Unter diese Kategorie fallen die großen Internet-Versandhändler. Das Geschäftsmodell funktioniert dann am besten, wenn man beim Verkaufen sein Wissen über den Kunden ausnutzt. Wie solche Anbieter gut mit Customer Relationship Management (CRM) umgehen können, kann man zum Beispiel sehr gut studieren, wenn man Kunde von amazon.com ist. Die Lesevorschläge, die man dort bekommt, stellen einen echten Mehrwert gegenüber einem normalen Einzelhandel dar. Die Auswahl ist enorm groß und die Lieferung ist nicht langsamer als bei einer Bestellung im normalen Buchhandel.
- Prozesseffizienz*     **Prozesseffizient, also B2C-Marktplätze und Vermittlung von Geschäften:** Eine weitere Kategorie von Geschäftsmodellen ist die Vermittlung von Geschäften. Darunter fallen zum Beispiel elektronische Auktionshäuser, elektronische Vergleichsplattformen und sonstige Vermittlung von Waren und Dienstleistungen über das Netz. Diese Geschäftsmodelle funktionieren, weil sie effizientere Prozesse bieten als ohne den Einsatz des Internets zur Unterstützung.

Allen den obigen Modellen ist gemeinsam, dass sie nicht unbedingt Treiber für EAI-Anwendungen sind. Am ehesten kann noch aus den letzten beiden Modellen der Bedarf nach EAI-Lösungen entstehen, nämlich dann, wenn für die Veranlassung von Lieferungen, Verbindungen mit Geschäftspartnern aufgebaut werden müssen.

Einige der aus den obigen Modellen hervorgegangenen Firmen sind stabil und werden bestehen bleiben. Viele weniger solide Marktteilnehmer mit schwachen Geschäftsmodellen sind schon ausgeschieden.

Übertragen auf Abbildung 4 befinden wir uns derzeit knapp hinter dem Punkt, der mit »The empire strikes back« markiert ist.



**Abbildung 4:**  
Hype-Kurve für E-Business und EAI  
Quelle: Generali eBusiness Strategieprojekt, Roland Berger Strategy Consultants, 2000

Dieser Punkt zeichnet sich dadurch aus, dass auch ältere, lange bestehende, konservative und etablierte Unternehmen anfangen, die Möglichkeiten des E-Business und des Internets zu erkennen und zu verwenden. Dort wird Internettechnologie vor allem für folgende Anwendungsfälle genutzt:

- **Neugestaltung von Geschäftsprozessen:** Begriffe wie Business Process Reengineering gibt es schon länger und die »radikale Neugestaltung von Geschäftsprozessen« hatte nicht immer einen guten Ruf. Trotzdem kann man in sehr vielen Unternehmen Kosteneffekte dadurch erzielen, dass man bestehende Geschäftsprozesse vereinfacht oder mit Hilfe von Technologieinsatz komplett neu erfindet. Dabei wird man vor allem versuchen, Bearbeitungsschritte zu automatisieren oder ganz wegfallen zu lassen und das 80/20-Prinzip anzuwenden. Also zum Beispiel 80% der Fälle möglichst automatisch abzuwickeln und menschliche Mitarbeiter nur dort einzusetzen, wo es unvermeidbar ist.
- **B2B-Integration:** Eine Variante der Neugestaltung von Geschäftsprozessen ist die Business-to-Business-Integration. Hier werden die Wertketten von Unternehmen durch Kommunikationstechnik verbunden. Es bestehen Beschleunigungs- und damit Kosteneffekte. Zum Beispiel kann das Bestellwesen automatisiert werden.

Reengineering von Prozessen

B2B-Prozess-automatisierung

- Selbstbedienung*  **Verlagerung von Service auf billigere Kanäle:** Zum Boom der Callcenter hat beigetragen, dass sie es erlauben, ähnlichen Service preiswerter zu erbringen als zum Beispiel mit einer flächendeckenden Service-Organisation. Selbstbedienung über das Internet ist noch mal preiswerter als Service über das Callcenter. Viele Unternehmen versuchen also möglichst viele Prozesse nach außen zu Kunden und Geschäftspartnern zu verlagern, um Kosten zu sparen.
- B2E-Portale*  **B2E-Modelle:** Eine Variante der Verlagerung von Service auf billigere Kanäle sind auch Business-to-Employee-Portale. Statt Mitarbeiter durch andere Mitarbeiter bedienen zu lassen, werden die nötigen Informationsleistungen über elektronische Portale erbracht.

Alle diese Ansätze haben eines gemeinsam: Um sie zu implementieren, muss man alte und neue Anwendungen integrieren und existierende DV-Systeme für Nachrichten aus dem Internet zugänglich machen. Dies sind also primäre Einsatzfelder für EAI-Technologien.

### 2.2.2 Trend 2: ERP-Lösungen und Komponenten

*ERP-Systeme* Ein weiterer Trend am Markt hat nur sekundär etwas mit dem Internet zu tun. So genannte ERP-Lösungen sind große fertige Softwarepakete, die von den meisten größeren Unternehmen heute für sekundäre Geschäftsprozesse wie Rechnungswesen, Materialwirtschaft und vieles mehr eingesetzt werden. Prominentester Vertreter solcher Systeme ist SAP.

Diese Systeme müssen natürlich mit den proprietären und zum Teil wettbewerbskritischen Anwendungen der Unternehmen verbunden werden.

Der Trend geht dahin, dass immer mehr Software in den Unternehmen im Einsatz ist und davon prozentual immer weniger selbst programmiert wird. Das heißt, dass sich die Aufgabenschwerpunkte von EDV-Mitarbeitern verschieben, und zwar von der Programmierung proprietärer Lösungen hin zur Integration fertiger Lösungen zu den Geschäftsprozessen, die das Unternehmen benötigt. Die EDV-Mitarbeiter der ERP-Anwenderunternehmen werden also nicht arbeitslos, sondern haben dann vor allem zwei Arten von Aufgaben:

- Customizing der zugekauften ERP-Anwendungen
- und Integration dieser Anwendungen mit den eigenen Anwendungen mittels GlueWare, wie zum Beispiel EAI-Tools.



Weitere Entwicklungen verstärken diesen Trend. Die Vorstellungen aus den 80er und 90er Jahren, dass ein Unternehmen ein möglichst integriertes Gesamtsystem in einer möglichst homogenen Technologie haben sollte, befinden sich auf dem Rückzug, weil sie sich als nicht praktikabel erwiesen haben.

- Zum einen sind die Technologiezyklen heute so kurz, dass es kaum noch ein Unternehmen schaffen wird, seine komplette Software innerhalb eines Technologiezyklus von 2–3 Jahren neu zu schreiben. Die typische Lebensdauer von geschäftskritischen Systemen liegt eher bei 20–30 Jahren und es dauert zum Beispiel in einer Versicherung um die 7–10 Jahre, ein Anwendungsportfolio einmal komplett zu erneuern. Daher müssen sich die Daten- und Architekturpolizisten mit dem Gedanken vertraut machen, es mit mehr als einer Technologiegeneration zu tun zu haben. Diese Systeme müssen integriert werden. Mithin handelt es sich wieder um einen potenziellen Anwendungsfall für EAI-Technologien.
- Zum anderen ist es gar nicht immer wirklich gut, ein voll integriertes System zu haben. Lose gekoppelte Komponenten können wesentlich leichter zu betreiben sein. Siehe dazu auch das Beispiel in Abschnitt 8.1.4, in dem es um die Vorteile »schlecht« integrierter Systeme geht.

*Kurze Technologiezyklen*

*Lose gekoppelte Systeme*

Schließlich gibt es noch den Trend »Componentware«. Wenn man Architekturen wie J2EE oder .NET als Componentware bezeichnet, wird Componentware eine Rolle spielen. Bei der Integration von existierenden, alten Systemen sind die Versprechungen des Component-Hypes allerdings nicht wahr geworden. Der wirklich große Component-Hype flaut daher schon wieder ab. Für EAI spielt Componentware nicht DIE große Rolle. Trotzdem werden wir auch die Integration über Komponenten in diesem Buch mit diskutieren. Siehe dazu Abschnitt 4.1.4.

*Componentware*

### 2.2.3 Trend 3: Fusionen

Eine weitere Strömung, die die Integration von Software zu einer häufig geübten Sportart macht, die jeder EDV-Profi beherrschen muss, sind die Unternehmenszusammenschlüsse und Fusionen. Sie kommen derzeit in der Finanzindustrie immer häufiger vor und haben teilweise gigantischen Umfang. Andere Industrien nutzen ebenfalls Economies of Scale.

*Fusionen*

Das Endziel bei solchen Fusionen wird es meist sein, das EDV-System eines der Fusionspartner zu behalten und die Bestände des anderen Partners in das beibehaltene System zu migrieren. Dieses Vorgehen ist

jedenfalls aus guten Gründen bei Fusionen im Finanzdienstleistungssektor sehr gebräuchlich, weil die Systeme meist skalierbar genug sind, um das Geschäft eines der Fusionspartner noch mit aufzunehmen. Man kann sich damit sehr oft doppelte Wartung ersparen<sup>2</sup>. Bis man so weit ist, vergehen allerdings meist 1,5 bis 3 Jahre, in denen man zum Beispiel gemeinsame Sichten auf Datenbestände beider Unternehmen benötigt. Dies kann dann wieder ein Anwendungsfall für EAI-Technologien sein.

## 2.3 Einige Anwendungsfälle für EAI-Lösungen

### *EAI-Anwendungsfälle*

In diesem Abschnitt lernen Sie vier typische Anwendungsfelder für EAI näher kennen. Die Beispiele stammen großteils aus dem Bereich der Finanzindustrie, denn dieses Buch hat, was die Beispiele anbelangt, einen Schwerpunkt auf Versicherungen und Banken. Der Analogieschluss zu anderen Industrien ist meist nicht sehr schwer.

Andere Fälle aus eher industriellen und sonstigen Umfeldern werden dann danach noch in Kürze vorgestellt.

### 2.3.1 Multichannel-Architekturen

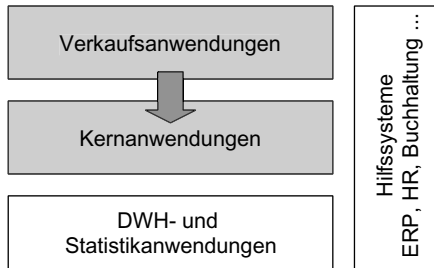
#### *Multichannel-Architekturen*

Um zu verstehen, warum man so genannte Multichannel- (oder auch Mehrkanal-) Architekturen unterstützen muss und welche Softwarewerkzeuge man dafür benötigt, ist ein kleiner Blick in die jüngere Vergangenheit von Versicherungen hilfreich.

Noch vor etwa 10 Jahren sah die Anwendungslandschaft eines Versicherungsunternehmens in etwa so aus, wie in Abbildung 5 dargestellt. Data-Warehouse-Anwendungen und ERP-Anwendungen waren oft noch in die Kernsysteme integriert. Uns interessiert für die Diskussion hier auch nur die Beziehung zwischen Verkaufssystemen (Front-Office) und Kernsystemen (Back-Office). Auch wenn im Vertrieb Laptops vorhanden waren, so erfolgte dennoch ein Großteil des Informationsflusses (Pfeil) zwischen Verkäufer und Back-Office über Papier.

---

2. Die Begründung dafür ist länger und nicht Thema dieses Buches und kann bei Interesse in [Kell2001] nachgelesen werden.

**Abbildung 5:**

Anwendungsportfolio grob – frühe 90er Jahre. Damals erfolgte ein Großteil des Informationsflusses zwischen Verkaufsanwendungen und Back-Office (Kernanwendungen) noch über Papier.

Schon damals existierte mehr als nur ein Vertriebskanal. So gab und gibt es als Vertriebskanäle den von Versicherungsunternehmen angebotenen Außendienst, Agenten, Makler und diverse Partnerorganisationen. Auch Kunden benötigen Auskünfte und direkte Kontakte.

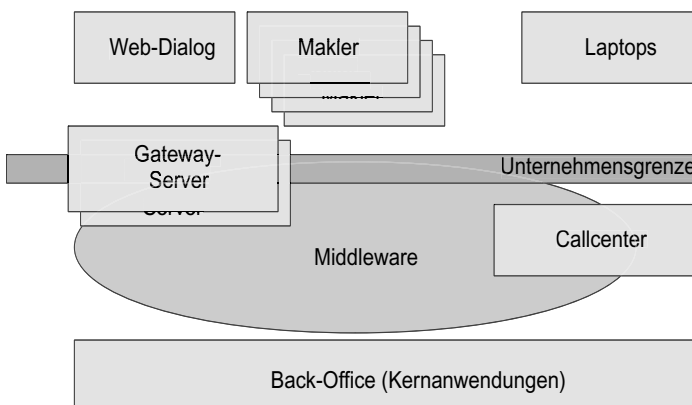
Mit dem Internet bieten sich nun mehrere Möglichkeiten der Verbilligung von Geschäftsprozessen an:

- Durch das Aufkommen von Callcentern und Internetdialogen konnte man darangehen, Auskünfte an Kunden auf Kanäle zu verlegen, die billiger sind als die Auskunft durch einen Außendienstmitarbeiter, Makler oder Innendienstmitarbeiter am Telefon.
- Die Außendienstkanäle, die nicht Teil des Unternehmens sind (Makler, Agenten, Partnerorganisationen), können ebenfalls mit Selbstbedienung und automatisiertem Datenaustausch besser und billiger angebunden werden, als das über telefonische Auskünfte durch einen Innendienst bisher der Fall war.

Callcenter

Außendienst

Diese beiden Entwicklungen hatten und haben Auswirkungen auf die Architektur der Informationssysteme. Die Welt vor dem Back-Office-System wird um einiges vielfältiger und komplexer.

**Abbildung 6:**

Typische Front-Office-Landschaft einer Versicherung

*Kanäle* Abbildung 6 zeigt dies. Vor der Unternehmensgrenze gibt es mehr als eine Anwendung, die Geschäftspartnern zur Verfügung steht. Für Kunden sind Web-Dialoge als Anwendungen zur Selbstbedienung vorhanden. Für Makler und Agenten gibt es entweder spezialisierte Anwendungen als Web-Anwendungen oder Datenaustausch über normierte Formate. Der eigene Außendienst arbeitet nach wie vor weiter mit Laptops oder demnächst permanent online mit Smartphones und Laptops. Das Callcenter steht als weiterer Kanal für die Bedienung von Kunden und Vertriebspartnern zur Verfügung.

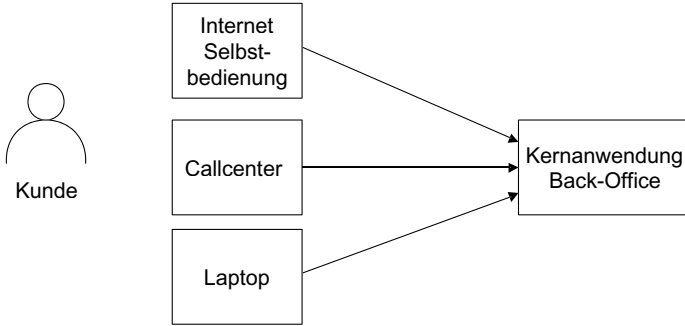
*Firewall* Die Unternehmensgrenze wird oft durch eine Firewallarchitektur dargestellt, durch die das Unternehmensnetz vom Internet abgetrennt und gegen Angriffe abgesichert wird. Gateway-Server übernehmen die Kommunikation nach außen.

Man kann dieses Bild sicher noch detaillierter darstellen. Wichtig ist jedoch vor allem der Aspekt der Middleware. Im Endeffekt werden bei der oben abgebildeten Multichannel-Architektur aus immer denselben Backend-Anwendungen die verschiedensten Frontend-Anwendungen bedient, allerdings ist die Aufbereitung jedes Mal eine geringfügig andere.

- Einige Kanalarten*
- ❑ Die Selbstbedienungs-Internetanwendung für Kunden lässt nur wenige, einfache Geschäftsfälle zu.
  - ❑ Im Kundenservicecenter (Callcenter) gibt es schon einen erweiterten Zugriff auf die Kernsysteme, zum Beispiel über einen Intranetdialog. Hier werden mehr Geschäftsfälle angeboten, zum Beispiel Änderungen von Verträgen.
  - ❑ Und die Offline-Laptops der Außendienstmitarbeiter werden über eine Datenversorgung bedient, die den Außendienst mit wichtigen Kunden- und Vertragsdaten beliefert. Es ist weiter damit zu rechnen, dass auch der eigene Außendienst Onlinezugriffe erhalten wird, wenn mobile Datenkommunikation so billig ist, dass das kostenmäßig verträglich ist. Es entsteht dann ein weiterer Onlinekanal.
  - ❑ Die gleichen Vertragsinhalte werden auch für Makler entweder im Dialog oder als Versorgungspakete für deren eigene Datenverarbeitung aufbereitet.

### **N:1-Multichannel-Architekturen**

Wir haben es also hier mit Architekturen zu tun, bei denen eine Art von Kunde über mehrere verschiedene Kanäle bedient wird. Allerdings werden die Kunden immer wieder aus demselben Backend-System mit grundsätzlich denselben Informationen beliefert. Abbildung 7 zeigt eine solche Situation. Ein Kunde kann hier über drei Kanäle bedient werden.

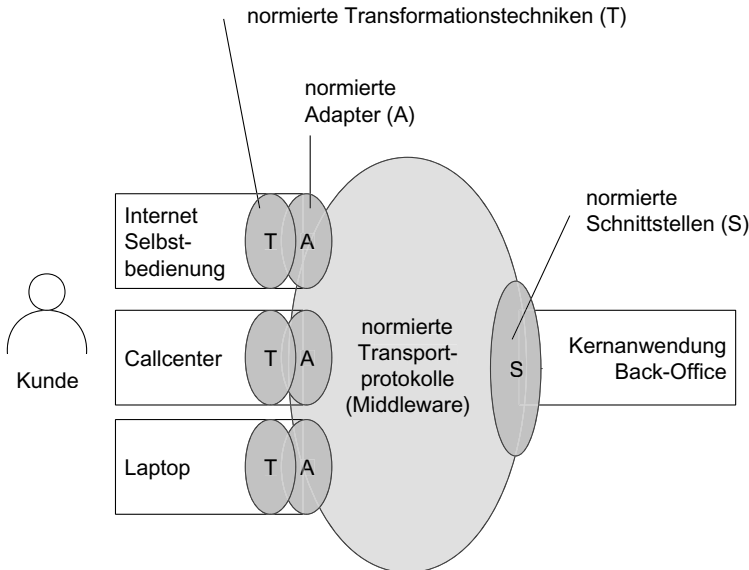


**Abbildung 7:**  
N:1-Multichannel-Situation

Abbildung 7 trägt aber bewusst noch nicht den Titel »Architektur«. Eine vernünftige Architektur muss es vermeiden, dass für jeden Kanal eine neue, individuelle Schnittstelle entsteht. Würde man ohne Architekturkonzept einen Kanal nach dem anderen an das existierende Kernsystem (Back-Office-System) andocken, könnte es zu einer Situation kommen, in der es für jeden Kanal eine eigene Schnittstelle für dieselbe oder sehr ähnliche Funktionalität gibt. Das wäre zu teuer und ist daher zu vermeiden.

Ein vernünftiger Einsatz von Middleware, und speziell von EAI-Middleware, erlaubt es hier, die Erstellungs- und Wartungskosten zu senken. Was für einen solchen Ansatz benötigt wird, ist abstrakt in Abbildung 8 gezeigt.

*Wartungskosten senken*



**Abbildung 8:**  
Einsatz von EAI-Technologien im N:1-Multichannel-Szenario

Einsparungen kann man durch folgende Vereinheitlichungen erzielen:

*Kostensenkung durch  
Vereinheitlichung*

- Zunächst macht es Sinn, wenn es möglichst nur eine Art der technischen Kommunikation gibt, damit nicht jeder Kanal über eine eigene Kommunikationsarchitektur mit der Kernanwendung kommuniziert.
- Dann sollte man die Schnittstelle des Kernsystems normieren. Man sollte versuchen, einfach aufzurufende Schnittstellen zur Verfügung zu stellen, die von mehr als einem Kanal genutzt werden können. Wer weiß, dass die Schnittstellen alter Hostsysteme vor allem für Onlinetransaktionen design sind, der weiß, dass die Normierung von Schnittstellen alles andere als ein triviales Unterfangen ist. Damit beschäftigen sich daher im Speziellen Abschnitt 4.2 und 8.2.1.
- Auch die Art, wie Frontend-Systeme auf die Middleware zugreifen, sollte weitgehend genormt sein. Wir sprechen hier im Folgenden von so genannten Clips. Siehe dazu speziell noch Abschnitt 5.1.2.
- Ebenso sollte auch die Art, wie aus einer möglichst gleichen Nachricht mehrere leicht unterschiedliche Darstellungen erzeugt werden, in einem Framework normiert sein. Die Werkzeuglandschaften rund um XML (wie zum Beispiel XSLT) bieten hierfür reichliche Möglichkeiten.

Wenn Sie EAI-Integrationsserver wie Vitria, TIBCO oder andere schon kennen sollten, werden Sie bemerken, dass man von den angebotenen Fähigkeiten solcher Tools für das hier dargestellte Szenario nur einen Bruchteil benötigt. Dies ist einer der Gründe, warum solche Tools nicht sehr verbreitet sind, wenn ein Kunde nur den oben dargestellten Anwendungsfall hat.

*Routing*

Speziell das so genannte Routing ist hier trivial. Unter Routing versteht man die Sicherstellung, dass ein Aufruf den richtigen Server erreicht. Wenn alle Aufrufe immer von demselben Backend bedient werden, ist das Routen der Aufrufe logischerweise trivial.

Wenden wir uns also einer etwas komplexeren Familie von Anwendungsszenarien zu.

### **N:M-Multichannel-Architekturen**

Bisher haben wir uns mit dem Szenario beschäftigt, bei dem im Wesentlichen ein Kernsystem (Back-Office-System) die Anforderungen aus den Kanälen abarbeitet. In den meisten Versicherungsunternehmen (und auch Banken) hat man es allerdings mit deutlich mehr als einem Server zu tun, der die nötigen Informationen liefert, um eine Gesamtansicht für einen Kunden zu erzeugen.

- Bei einer Versicherung sind typischerweise oft das Sachversicherungssystem und das Lebensversicherungssystem getrennt. Häufig kann es auch mehr als ein Lebensversicherungs- oder Sachversicherungssystem geben.
- In einer Bank gibt es typischerweise mindestens ein System für laufende Konten und eines für Wertpapierhandel.

Oft liegen alle diese Systeme auf einer Maschine. Manchmal ist aber auch das schon nicht mehr der Fall. Spätestens dann aber, wenn ein Unternehmen auch Marktleistungen anderer Anbieter offeriert, ist das N:1-Multichannel-Szenario durchbrochen. Sowohl Banken als auch Versicherungen verkaufen Produkte des jeweils anderen Bereiches der Finanzindustrie mit und versuchen Allfinanzkonzepte zu implementieren.

*N:M-Multichannel*

Damit ergibt sich ein Trend zu mehr Kanälen und gleichzeitig mehr Servern. Diese Server für Kernapplikationen (Back-Office-Systeme) werden wir im Folgenden Produktfabriken nennen. Wir haben es durch Ideen zur Kostensenkung mit mehreren Kanälen zu tun. Durch die horizontale Integration von neuen Produkten werden gleichzeitig im Bereich der Back-Office-Systeme mehrere Produktfabriken angesprochen. Um zum Beispiel als Versicherung ein Allfinanzangebot zur Verfügung stellen zu können, benötigt man noch mehr Server als Versicherungsvertragssysteme (siehe Abbildung 9), nämlich Produktfabriken für Bankprodukte.

*Viele Server*



**Abbildung 9:**  
Multichannel-Architektur  
und integrierende  
Middleware

Abbildung 9 zeigt, dass man, um zum Beispiel ein einheitliches Web-Interface für einen Kunden anzubieten, auf Versicherungsvertragssysteme zugreifen muss oder nun auch zum Beispiel auf eine Girokonto-Maschine.

Gleichzeitig müssen die Informationen der Sachversicherungsmaschine (Vertragssystem) in mehr als einer Oberfläche angezeigt werden.

Solche Probleme geht man typischerweise wieder mit EAI (Enterprise Application Integration)-Lösungen an.

**Abbildung 10:**

Einsatz von  
EAI-Technologien im  
N:M-Multichannel-  
Szenario

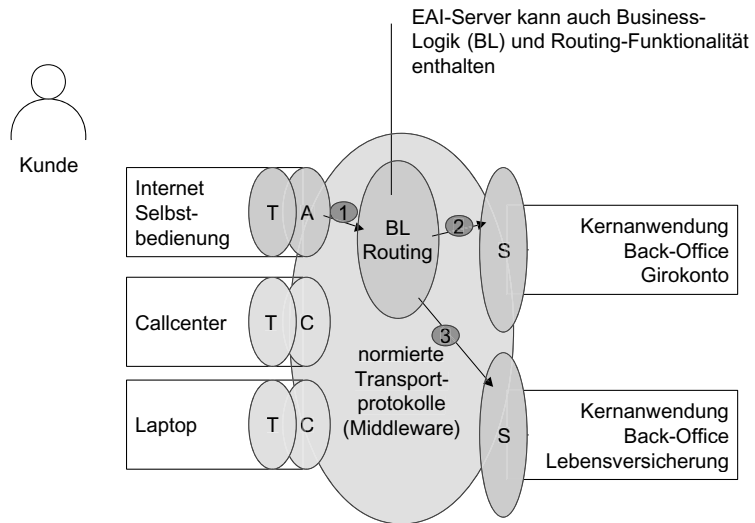


Abbildung 10 ist geringfügig komplexer als Abbildung 8. Als zusätzliche Funktionalitäten sollte die Middleware praktischerweise noch Folgendes beherrschen:

*Business-Logik*

- Es sollte möglich sein, mit Hilfe der Business-Logik Dienste auf Ebene der Middleware zu definieren, die es in keiner der angeschlossenen Produktfabriken schon gibt. Zum Beispiel könnte sich der »Netto-Vermögenswert« eines Kunden aus dem Wert seiner Girokonten und Lebensversicherungen zusammensetzen. Es macht keinen Sinn, wenn jeder Channel diese Formel und dieses Wissen wieder neu implementiert. Besser ist es, wenn dies zum Beispiel in einem EAI-Server implementiert wird. BL steht dabei in Abbildung 10 für Business Logic, also Geschäftsfunktionalität.

*Routing*

- Weiter macht es Sinn, zumindest elementare Routing-Fähigkeiten in der Middleware zu haben, da man nicht jede Menge Software durchsehen möchte, wenn sich ein Produktgeber oder auch nur die Adresse der Produktfabriken im Netz ändert.

### Lage eines EAI-Servers in einer N:M-Multichannel-Architektur

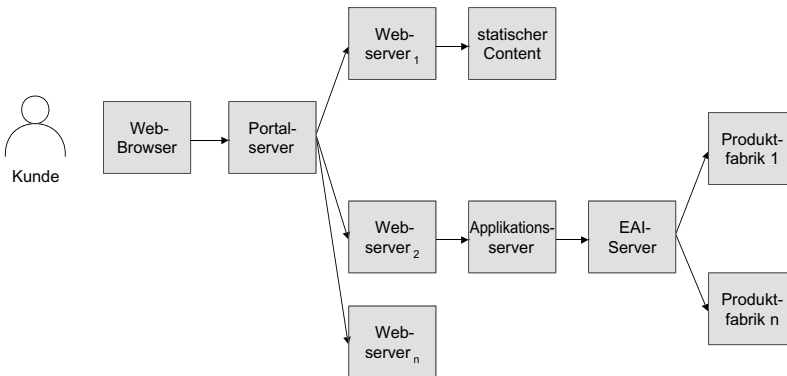
In Abbildung 8 und Abbildung 10 wurde die EAI-Funktionalität als Wolke zwischen den Kanalanwendungen (Clients, Frontends) und den Produktfabriken gezeichnet. Eine solche Wolke gibt es weder als »Stück Software« noch wäre dies eine besonders befriedigende Archi-



tekturskizze. Wir wollen daher kurz beschreiben, mit welcher Art von Servern man es dann in der Realität zu tun hat und welche Verantwortlichkeiten diese haben. Nicht jeder Server muss auf einer eigenen Maschine laufen, er könnte es aber.

Für die Beispiele und Arten von Servern und Software werden wir im Folgenden die Terminologie von J2EE (Java 2 Enterprise Edition) verwenden. Analoge Architekturen kann man auch für Microsofts .NET-Plattform aufbauen, allerdings mit einer leicht unterschiedlichen Terminologie.

J2EE und EAI



**Abbildung 11:**  
Vom Browser zur  
Produktfabrik

Die in Abbildung 11 dargestellten Server haben die folgenden Verantwortlichkeiten:

er er	er nt rt n
Portalserver	Der Portalserver bietet dem Benutzer einen Einstieg (auch personalisiert), die Authentifizierung und bindet statische Web-Inhalte an.
Webserver	<p>Die Webserver stehen hier für Server für statische oder dynamische Inhalte.</p> <p>Die Webserver können auch Komponenten enthalten, die Benutzerinteraktionen auf der Basis von Eingaben des Benutzers steuern. Das geht dann über die reine statische Web-Interaktion hinaus. Die Formulare einer Anwendung für Überweisungen würden zum Beispiel von einem Webserver gerendert werden. Dieser würde auch einen Session-Kontext für den Benutzer mindestens während seiner Überweisung halten.</p> <p>Im J2EE-Modell würden auf dem Webserver vor allem die JSPs (Java Server Pages) laufen, die die dynamischen Webseiten erzeugen.</p> <p>Es ist klar, dass hinter einem Portalserver mehrere Applikationen und damit beliebig viele Webserver in Frames laufen können.</p>

er er	er nt rt n
Applikationsserver	Ein Applikationsserver stellt Geschäftsfunktionalität zur Verfügung. Der Webserver kann entweder einen Applikationsserver benutzen, wenn der Kunde komplexere Objekte zu sehen bekommt. In einem solchen Fall würde auf dem Applikationsserver zum Beispiel ein EJB-Container laufen. Oder es können auch Proxies (Java-Servlets) für andere Funktionalitäten dort laufen.
EAI-Server	Der EAI-Server übernimmt Business Logic und Routing sowie unter Umständen auch Schnittstellennormalisierung für nachgelagerte Back-Office-Systeme. Die genaue Aufgabenteilung zwischen EAI-Server und Back-Office-System wird in Kapitel 5 noch am Beispiel eines EAI-Servers exemplarisch erläutert werden.
Backend	Die Backend- oder Produktfabriken erbringen Dienste für den Benutzer.

Die Architektur in Abbildung 11 ist in dieser Form noch sehr abstrakt dargestellt. Es gibt durchaus noch Designfragen, die gestellt werden und für die man eine für das jeweilige Projekt angemessene Antwort finden muss:

*Portalserver*

- Portalserver sind noch nicht sehr weit verbreitet. Sie können auch durch einen Applikationsserver mit starker Personalisierung abgebildet oder einfach weggelassen werden.

*Integration über Frames*

- Web-Anwendungen können über Frames aus Anwendungen mehrerer Webserver zusammengesetzt werden. Die Frage, ob man fremde Teile und Unterapplikationen einfach über einen Frame einblendet oder die Reaktionen des Benutzers selbst kontrolliert, kann eine sehr entscheidende für das Customer Relationship Management sein. Im einen Fall weiß der Informationslieferant für einen bestimmten Inhalt, wohin Ihr Benutzer klickt. Wenn Sie sich die Mühe machen, die XML-Datenströme des Informationslieferanten selbst aufzubereiten und darzustellen, wissen Sie auch selbst, ob Ihr Kunde sich mehr für riskante oder konservative Aktien interessiert.
- Hinter einem Webserver können beliebig viele Applikationen und Applikationsserver stehen. Deren Partitionierung ist eine Designfrage.
- Die Mächtigkeit des eingesetzten EAI-Servers und die Kommunikationsmodelle sind ebenfalls interessante Designfragen, die von Anforderungen getrieben werden.
- In Abschnitt 4.2 und 5.1.4 werden wir sehen, dass es auch Unterschiede macht, wie man das Thema der Schnittstellennormalisierung angeht und ob man Gateways oder Wrapper verwendet.

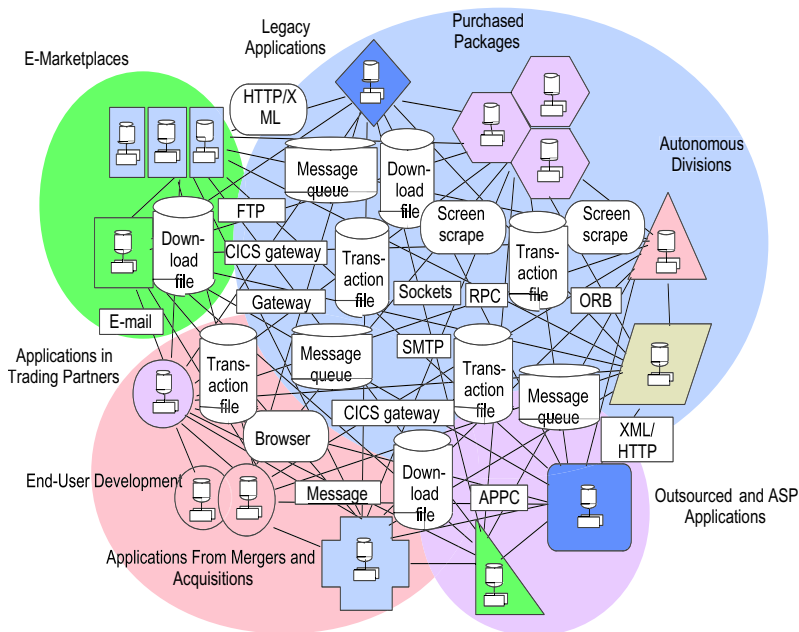
Von daher kann Abbildung 11 wirklich nur eine grobe Orientierung geben. Eine Architektur ist damit noch lange nicht festgelegt. Dazu benötigt man zum Beispiel noch die genauere Beschreibung verschiedener Architektursichten. Eine vertiefende Anleitung dazu finden Sie zum Beispiel in [Star2002].

### 2.3.2 A2A-Kommunikation

Eine der verbreiteten Geschichten, die benutzt werden, um EAI zu verkaufen, ist die so genannte A2A (Application to Application)-Integration. Der Anwendungsfall liegt dann vor, wenn ein Unternehmen viele Inselanwendungen hat, die auch auf sehr vielen Servern liegen können und die miteinander kommunizieren müssen. Abbildung 12 zeigt viele Anwendungen, die über Punkt-zu-Punkt (P2P)-Verbindungen miteinander kommunizieren.

A2A

P2P



**Abbildung 12:**  
Vermeintliches Kommunikationschaos auch bekannt als Spaghetti-Kommunikationsarchitektur  
(Quelle Gartner Research, Massimo Pezzini)

Es ist nun leicht ersichtlich, dass Wartungskosten potenziell hoch sind, wenn man es mit vielen Verbindungen und Schnittstellen zu tun hat, die alle nach unterschiedlichen Standards und mit unterschiedlichen Werkzeugen programmiert wurden.

Theoretisch kann man es mit der Ordnung von  $O(n^2)$  Verbindungen zu tun haben, wenn jede Anwendung mit fast jeder anderen kom-

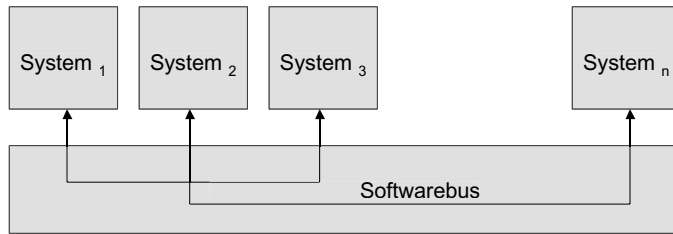
Problem der  $O(n^2)$   
Verbindungen

muniziert. Wir werden allerdings sehen, dass dies bei Versicherungen und Banken so gut wie nicht in dieser Schärfe vorkommt (siehe dazu dann Abschnitt 2.3.5).

*Softwarebus*

Um das  $O(n^2)$ -Kommunikationschaos in den Griff zu bekommen, führen A2A-Kommunikationslösungen einen Softwarebus oder einen Sternverteiler ein. Dadurch erreicht man, dass aus  $O(n^2)$  Interfaces nur noch  $O(n)$  Adapter werden.

**Abbildung 13:**  
Softwarebus –  
analog CORBA



Allerdings haben wir hier durch die Einführung einer Kommunikationsinfrastruktur ebenfalls bei weitem noch nicht alle Probleme gelöst:

*Verbleibende Probleme*

- ❑ Die Normalisierung von Schnittstellen wird einem durch die Infrastrukturen nicht abgenommen. Oft sind Schnittstellen auch deshalb schlecht zu warten, weil sie alt, schwer zu verstehen und eben nicht normalisiert sind. Das Reengineering auf der Ebene von Code, das erforderlich ist, um die Schnittstellen zu verbessern, wird von keinem Tool übernommen.
- ❑ Der Aufwand, alle Systeme an einen Softwarebus anzudocken, darf nicht unterschätzt werden.
- ❑ Oft kommunizieren Systeme auch über Dateien. Wenn man dies auf aufruforientierte Schnittstellen von Softwarebus-Infrastrukturen umstellen möchte, ist dies mit erheblichem Aufwand verbunden.

*EAI als Softwarebus*

Wenn man sich entscheidet, eine EAI-Lösung als Softwarebus einzusetzen, hat man sehr viele Produktalternativen, die sich in ihrer Mächtigkeit erheblich unterscheiden. Hier seien nur einige genannt, die teilweise im Folgenden noch näher erklärt und abgeglichen werden:

*DCE*

- ❑ DCE (Distributed Computing Environment) [IBM2001] ist eine Umgebung, die es erlaubt, auf einheitliche Weise so genannte Remote Procedure Calls auszuführen. DCE ist aber gerade in alten Hostumgebungen nicht für alle Programmiersprachen und Betriebssysteme verfügbar, sondern eher in Unix-Umfeldern beheimatet. Außerdem unterstützt DCE vor allem synchrone Prozeduraufrufe.

- CORBA (Common Object Request Broker Architecture) ist eine modernere Umgebung, die als Bus für objektorientierte Programmierung gedacht ist. Wie DCE ist auch CORBA eher für »modernere« Umgebungen geschrieben und tut sich mit der Einbindung älterer Umgebungen schwer. Wenn man zum Beispiel alte CICS- und IMS-Programme einbinden möchte, benötigt man zusätzliche Gateways.

*CORBA*
- Diverse EAI-Server umgehen die oben genannten Probleme, indem sie eine Vielzahl von Adaptern zur Verfügung stellen und darüber hinaus Adaptergeneratoren für ältere 3GL-Sprachen wie COBOL und PL/I anbieten.

*EA-Server*

### 2.3.3 Geschäftsprozessintegration

Das Szenario, dem EAI seinen Namen eigentlich verdankt, ist die Geschäftsprozessintegration. Industrieunternehmen waren und sind häufig mit der Situation konfrontiert, dass diverse Systeme integriert werden müssen, um einen Geschäftsprozess zu steuern:

*Integrierte  
Geschäftsprozesse*

- In einem Verkaufssystem werden Aufträge erfasst.
- Dies führt zu einer Lagerprüfung und zum Beispiel, falls der Artikel nachproduziert werden muss, zu einem Produktionsauftrag an ein Produktionsplanungs- und Steuerungssystem.
- Dieses System muss selbst wieder Lagerprüfungen und Beschaffungen auslösen. Letztere laufen in ein Einkaufssystem.
- Der Produktionsauftrag wird unter Umständen, wenn alle Vormaterialien vorhanden sind, an ein Fertigungssteuerungssystem weitergegeben, das Maschinen oder Produktionseinrichtungen steuert.

Große Teile solcher Prozesse sind zwar zum Beispiel durch integrierte ERP-Pakete, wie SAP, Baan, Oracle oder Peoplesoft, abgedeckt. Es können aber sicher nicht alle Funktionen in einem Unternehmen durch solche Pakete ausgeführt werden. Man bekommt also Integrationsprobleme, dadurch dass aus einem Geschäftsprozess viele verschiedene Systeme angesteuert werden müssen.

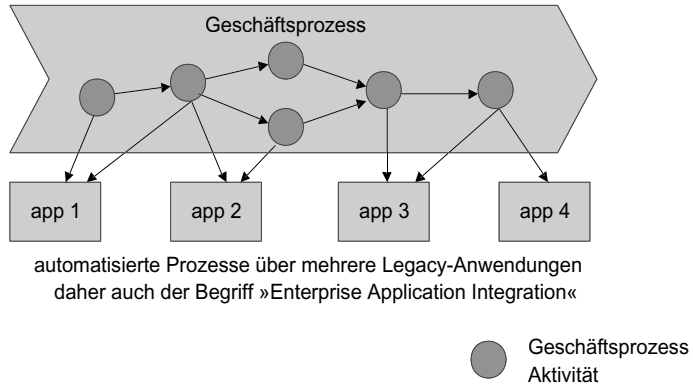
*Integrierte Prozesse und  
ERP-Pakete*

Abbildung 14 zeigt eine solche Lösung schematisch. An eine steuernde Applikation (in diesem Fall dargestellt durch den Geschäftsprozess) müssen über Schnittstellen viele andere Anwendungen angebunden werden. Die Arten dieser Anbindungen können dabei sehr verschieden sein:

- Es kann sein, dass aus dem Geschäftsprozess ein Bildschirm einer anderen Anwendung geöffnet wird, der Benutzer darin ein paar Eingaben macht und dann im Geschäftsprozess fortfährt. Dies

*Integration über die  
Oberfläche*

**Abbildung 14:**  
Intention von Enterprise  
Application Integration



wird als Integration über die Benutzungsoberfläche bezeichnet (siehe dazu auch Abschnitt 4.1.1).

*Funktionale Integration*

- Es ist auch möglich, dass der Geschäftsprozess synchron oder asynchron Funktionalitäten der anderen Systeme aufruft und dafür eine eigene Benutzungsoberfläche verwendet hat. Die verschiedenen Arten der Integration über Funktionsaufrufe sind in Abschnitt 4.1.2 näher beschrieben.

*Integration über Daten*

- Oder aber, der Geschäftsprozess kann auch über Datenbankzugriffe und Dateien mit den Anwendungen kommunizieren, auch wenn davon eher abzuraten ist (siehe Abschnitt 4.1.3).

In jedem der Fälle gibt es eine »steuernde Anwendung«. Dies kann zum Beispiel ein Geschäftsprozessmanagementsystem sein, wie IBM MQSeries Workflow<sup>3</sup>.

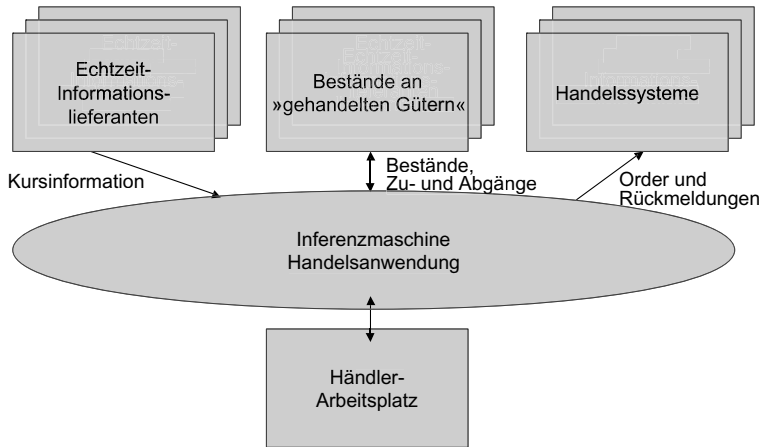
### 2.3.4 Trade Rooms

*Händlerarbeitsplätze*

Trade Rooms sind Arbeitsplätze für Händler. Trade-Room-Arbeitsplätze gibt es zum Beispiel für Devisenhändler, Aktienhändler, Derivatgeschäfte aller Art oder Warentermingeschäfte. Trade Rooms sind ebenfalls ein lohnendes Einsatzgebiet für EAI-Technologien.

Einige Integration Broker kommen speziell aus dem Gebiet der Trade Rooms. Abbildung 15 zeigt grob, was es dort zu integrieren gibt:

3. In Zukunft WebSphere Workflow



**Abbildung 15:**  
Stark schematisierte  
Trade-Room-Anwendung

Ein Händler bezieht seine Informationen aus einem oder mehreren Echtzeit-Informationssystemen, die Kursinformationen von den Handelsplätzen liefern. Diese Informationen können dazu führen, dass der Händler Handelsaufträge auslöst. Diese können auf mehreren Handelsplätzen abgewickelt werden – was normalerweise dann auch in verschiedenen Handelssystemen passiert. Durch die Rückmeldungen werden die eigenen Bestände verändert.

*Blackboard-Architektur*

- ❑ Es handelt sich bei Handelsarbeitsplätzen um Echtzeitanwendungen, die den Benutzer in die Lage versetzen müssen, innerhalb sehr kurzer Zeitspannen reagieren zu können.
- ❑ Aus den Strömen von Kursdaten müssen die relevanten herausgefiltert werden und in sinnvolle Ereignisse umgesetzt werden.
- ❑ Es werden simultan mehrere Ströme von Kursdaten angeliefert.
- ❑ Es können Aufträge an verschiedene Handelssysteme ergehen mit jeweils anderen Schnittstellenkonventionen.
- ❑ Und auch die asynchronen Rückmeldungen der Handelssysteme müssen verarbeitet werden und führen zu Events.
- ❑ Dieser Abschnitt kann keinen tieferen Einblick in die Geschäftslogik von Handelssystemen geben. Es soll nur klar gemacht werden, dass eine solche Anwendung hohe Anforderungen an die Integration der beteiligten Softwaresysteme stellt.

Anwendungen dieser Art sind daher eine Domäne für spezielle Integrationsserver, wie TIBCO, die mit für Trade Rooms entwickelt wurden.

### 2.3.5 Was ist bei Banken und Versicherungen anders als bei der »klassischen EAI-Story«?

Speziell die A2A-Geschichte wird sehr häufig von Verkäufern von EAI-Produkten genutzt, um auf Vorstandsebene Integrationsserver zu verkaufen. Die meisten Chief Information Officers haben das Problem, Geschäftsprozesse schnell anpassen zu müssen. Sie haben dabei nicht die Zeit, um Anwendungen komplett neu zu bauen. Sie müssen daher meist viele existierende Anwendungen und wenige neue Anwendungen zu immer neuen Prozessen zusammenbauen.

*CIOs müssen Prozesse  
schnell anpassen*

Vor solchen Herausforderungen stehen derzeit auch viele Banken und Versicherer und sie bekommen als Lösung so genannte Integrationsserver angeboten. Die Geschichte, mit denen die Verkäufer dieser Lösungen antreten, ist oft geradezu ein Horrorszenario. Abbildung 12 hat ein solches Szenario gezeigt, in dem jede Anwendung mit jeder kommunizieren muss. In der dabei entstehenden »Sprachverwirrung« steigen die Wartungskosten für die Integration von Schnittstellen ins Unermessliche. Dies ist der benötigte Nutzen, mit dem man die Investitionen in ein Off-the-shelf-EAI-Tool rechtfertigen kann, und zwar auch dann, wenn sie hoch sind.

Es gibt zwei Szenarien, mit denen am meisten argumentiert wird:

- Das A2A-Szenario, das in Abschnitt 2.3.2 beschrieben wurde,
- und das Geschäftsprozessintegrationsszenario aus Abschnitt 2.3.3. Letzteres wird allerdings schon weniger herangezogen.

*Multichannel*

Für Versicherungen und auch für Banken ist jedoch aus unserer Sicht das Multichannel-Szenario einer der wesentlichen Treiber für die Einführung von EAI-Werkzeugen. Dieses wird allerdings nur wenig »bevorzogen«.

Das A2A- oder Softwarebus-Szenario ist aus mehreren Gründen nicht das typische Szenario:

*Versicherungssysteme  
Bankssysteme*

- Versicherungen oder Banken haben meist ziemlich große Einzelsysteme, wie zum Beispiel ein komplettes Lebensversicherungssystem, die in sich integriert sind. Dass ein Lebensversicherungssystem noch mit vielen weiteren Servern kommuniziert, die nicht auch im selben Adressraum mit integriert sind, ist schon aus Performanzgründen sehr selten. Vielmehr reicht zum Beispiel das Lebensbestandssystem einer Versicherung aus, um komplette Geschäftsfälle (also Geschäftsprozessinstanzen) wie einen Neuantrag oder eine Vertragsänderung abzuwickeln. Daher trifft auch das Geschäftsprozessintegrationsszenario kaum jemals zu.

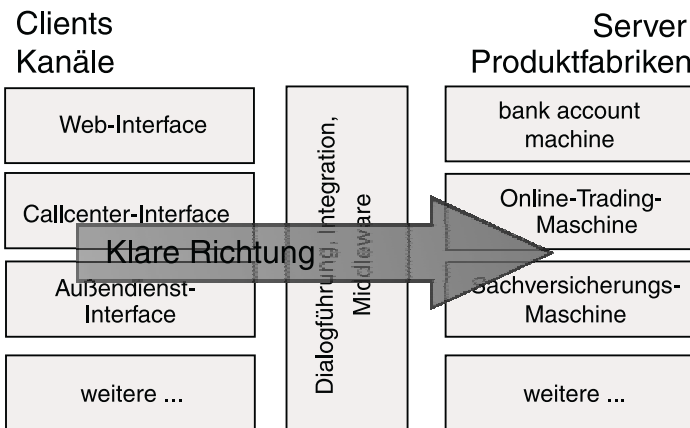


- Versicherungen und Banken haben keine Zulieferer und Warenwirtschaft existiert nur in Nebenbereichen. In den Kernprozessen gibt es keine Warenwirtschaft<sup>4</sup>, da das Produkt »pure Information« ist.
- Echte verteilte Verarbeitung, indem jedes System mit jedem kommuniziert, ist unerwünscht, weil teuer. Im Gegensatz zu dem allgerneinsten A2A-Szenario haben Banken und Versicherungen in einem Multichannel-Szenario eine klare Richtung in ihrer Kommunikation vom Kunden zu den Back-Office-Systemen und zurück (siehe Abbildung 16). Weiter kommt dort vereinfachend hinzu, dass fast alle unsere Server auf OS/390-Maschinen liegen und dort auch noch länger bleiben werden und die meisten Clients über Web (Applikationsserver) bedient werden.

*Fehlende Warenströme*

*Verteilte Verarbeitung ist komplex*

Das Trade-Room-Szenario ist zwar auch interessant, ist aber auf einige Inseln in einer Architektur beschränkt. Damit ist für Versicherungen und auch für Banken aus unserer Sicht das Multichannel-Szenario das interessanteste Szenario. Aus diesem Blickwinkel wollen wir daher im Folgenden vor allem EAI-Werkzeuge betrachten.



**Abbildung 16:**

*Kommunikation hat bei Banken und Versicherungen meistens eine klare Richtung*

4. Lässt man bei Banken die Verwaltung der physischen Wertpapiere und Geldmengen einmal außer Acht. Versicherungen haben auch dies nicht.